# CHECKLIST USED FOR EDA & DEVELOPING LINEAR REGRESSION MODELS WITH NEW DATA

SimpleAI.ch

## - Step 1 - DATA INGESTION

- Load & merge data into one dataframe with all features and representative number of examples
- Check df shape, indexes, duplicated cols & rows, and if you created new rows/columns with NA's

### Rows/Col with too many NA

Optionally, Remove/or label samples/rows & columns with too many NA, or only NA

### Anomaly Detection

Do you see any samples with more then average number of zeros/NAs/anomalies/novelties

## - Step 2 - DATASET SUMMARY

**Dataset Overview**
- What is in my data?
- How many variables/samples?
- What are feature dtypes, and are they correctly encoded?

**Missing Data**
- How much of NA do you have in the dataset - per row/col, and eg: how many rows/col have >=90% of missing data
- What is the detailed distribution of missing data ie. which rows/columns
- What other values can be interpreted as NA, eg: 0, or "no value" text
- what NA, can be encoded as 0, or class/category

**Feature Values**
- what is the range or class number in each feature?
- Which var. have imbalanced class/value distribution,
- Get examples of each feature
- Are there any obvious OUTLIERS?
- How many zero values are there,

**Check for spacial/datetime variables**
- Check datetime features
- Identify columns with ID, sample order, or similar, and check if they are all unique

## - Step 3 - DATA CLEANING

### 1. Define Variable Groups

(target, quantitative, ordinal, nominal, datetime, spatial, ID columns)

### 2 Cleaning & Organisation

**Format dtypes**
- Set proper dtype for each feature
- Make sure date-time variables are properly formatted,
- If exported from Excel (at any point) check for numerical values turned into data-time

**Remove Duplicates**
- Identify and remove any duplicates (rows & columns)

**Missing data**
- Check if all missing data are encoded in the same way, if not use one notation,,
- No imputation at this stage !
- The goals, is to have all missing data encoded as NA, and not something else or to remove, rows/columns that have only missing data

## - Step 4 - ENCODING

### Define Encoding policy

*The goal is to have consistent, and non-ambiguous encoding in entire dataset*

**For quantitative variables**
- Identify incorrect values: Eg: -100 foot size, (plots, histograms, get examples)
- Next, prepare dict./note/script with (i) feature name, (ii) incorrect values, and (iii) actions to take to either Remove, Replace or clip identified incorrect values.

**For ordinal variables**
- Define how many classes, and how should they be called
- To help, checking for correlations, replace class names with sorted integers, means or medians (target var), and store it in a new column

**For nominal Variables**
- text/object; Tokenize text/object values or replace with classes
- Define proper encoding for each class Eg: remove inconsistent class names such as "House"/"Home" -> "Home"

**For datetime Variables**
- Make sure date-time variables are properly formatted, (one timezone)
- Encode these variables, eg: age, years/days since last renovation, etc...

## - Step 5 - CHECK MODEL ASSUMPTIONS

*TIP: transform nominal var's into ordinal using the target var. to include them in the analysis*

### Check Target Variable
- Does it have Normal distribution?
- Use: histogram, boxlot, QQ-plot + stat tests + skewness, kurtosis, mean, median
- Check if scaling, or log-transformation should be used to improve residuals disrib.

### Perform Correlation analysis

***- Pair-wise comparisons -***
- First. Focus on Xi~dependent variable , to identify predictors correlated with the target variable (use: barplot, table)
- Then, analyse correlations between each variable pair, to find collinear var's that should be removed (use: annotated heatmap & table with sorted results)

***- one vs rest comparison -***
- Calculate ViF score for each feature, to detect potential muticolinearity (use: barplot, & table to present the results)

### Check Quantitative variables

**- Distribution -**
- Are they normally distributed?
- If not, try transformations for each var. eg: log, sqrt, polynomial. Etc... list propositions,

**- Extreme Values -**
- Can you find outliers or groups of outliers/zeros in the data? (eg. using z-scores, on scatter plots)
- Do you see any samples with more then average number of zeros, Nas, anomalies, or novelties

### Check linear model validity
- Main question: can we fot linear regression model with our data?
- Plot target variable against any predictor. Use, scatter plot with smoothed trend-line, or boxplots. Add info on correlation, NA%, mean, median, sd, to each plot,

## - Step 6 - FEATURE ENGINEERING & F. SELECTION

**GOAL: To generate 2D matrix with only numerical data (int, floats, binary). The selected, & transformed features should have following characteristics:**

(i) No autococelation/batch effects
(ii) linear relationship with the target variable
(iii) multivariate normal distribution
(iv) Scaled or normalized values
(v) No missing data
(vi) Best, No outliers, or identified outliers, + vector with weight

***use Sklearn functions to create transformers, and add them to one piepline***

**DEFINE**
- Features that you wish to use in the model (see feature selection part in Notebook 2)
- transformers for each variable
- imputation strategy
- variables for one-hot encoding
- numerical variables that wiii be scaled, standardized of normalized, or other...

**CHECK**
- How does your transformers treat novelties, especially for nominal variables,

**REMEMBER TO SAFE**
- Feature names, and sample ID for each cell/row/column
- Dataset & pipeline version
- Source data


See Notebook 2

## - Step 7 - MODEL TRAINING & EVALUTATION

Here we only establish a simple pipeline for testing assumptions and setting up automated functions

See Notebook 2

## - Step 8 - ERROR ANALYSIS

### 1. Model performance assessment

- Bar-plot, or boxplot with errors calculated for the model/s and baseline (*RMSE, MAE, R^2*) to select best performing models
- Add test a& training error
- Print table with models sorted by test error, with information on hyperparameters and data transformations and features

### 2. Residual analysis

*- Done for selected models -*

**Create Four Standard Plots**
- **Plot 1. Histogram**
- **Plot 2. QQ-Plots** or **PP-Plots**
- **Plot 3. residuals vs target variable**
- **Plot 4. Residuals vs sample order/ID**

**Test Residuals Normality**
- Use plot 1 & 2
- Mean; Median, Skewness; Kurtosis;
- Shapiro–Wilk test, Scipy normaltest, Lilliefors or Kolmogorov-Smirnov

**Test Residuals Homoscedascity**
- Use plot 3 - fannning effect
- White test; or Breusch-Pagan test

**Test for residuals autocorelation**
- Use plot 4 – look for patterns
- Durbin-Watson Test

### Step 3. Check for latent variables

- check if residuals/ target values are correlated with (i) amount of missing data/zeros per sample, (ii) sample order (plot 4) (iii) or the % of extreme values in each sample
- + Look for batch effects & autocorrelation,
- Then, use PCA or t-SNE to check if you can cluster the samples with any spatial var's

### *Additional Analyses:*

- ***Model Resilance;*** *how much of missing data, novelties, or values outside feature space the model can tolerate*
- **Feature importance criterion**
- **Results stratification**
- ***Validation Dataset*** *-. used to test the final model*